

```
(*
      CS 51 Lab 15
      Native Lazy Streams
*)

(*.....
An implementation of lazy streams using OCaml's native 'Lazy' module,
along with with some useful functions.

*****
YOU SHOULD NOT EDIT THIS FILE.
*****

*)

type 'a stream_internal = Cons of 'a * 'a stream
and 'a stream = 'a stream_internal Lazy.t ;;

(* head strm -- Returns the first element of 'strm'. *)
val head : 'a stream -> 'a ;;

(* tail strm -- Returns a stream containing the remaining elements of
'strm'. *)
val tail : 'a stream -> 'a stream ;;

(* first n strm -- Returns a list containing the first 'n' elements
of the 'strm'. *)
val first : int -> 'a stream -> 'a list ;;

(* smap fn strm -- Returns a stream that applies the 'fn' to each
element of 'strm'. *)
val smap : ('a -> 'b) -> 'a stream -> 'b stream ;;

(* smap2 fn strm1 strm2 -- Returns a stream that applies the 'fn' to
corresponding elements of 'strm1' and 'strm2'. *)
val smap2 : ('a -> 'b -> 'c) -> 'a stream -> 'b stream -> 'c stream ;;

(* sfilter condition strm -- Returns a stream of all elements of
'strm' for which 'condition' holds. *)
val sfilter : ('a -> bool) -> 'a stream -> 'a stream ;;
```